**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

| | |
|---|---|
| In re Application of: | |
| Inventors: Guogen Zhang et al. | Examiner: Charles D. Adams |
| Serial #: 10/629,459 | Group Art Unit: 2164 |
| Filed: July 29, 2003 | Appeal No.: _________ |
| Title: DYNAMIC SELECTION OF OPTIMAL GROUPING SEQUENCE AT RUNTIME FOR GROUPING SETS, ROLLUP AND CUBE OPERATIONS IN SQL QUERY PROCESSING | |

REPLY BRIEF OF APPELLANTS

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 C.F.R. §41.41, Appellant's attorney hereby submits the Reply Brief of Appellants in response to the Examiner's Answer dated April 1, 2009 for the above-identified application.

No fee is required for filing this Reply Brief. However, the Office is authorized to charge any necessary fees or credit any overpayments to Deposit Account No. 09-0460 of IBM Corporation, the assignee of the present application.

## I. ARGUMENTS

In the Answer, the Examiner essentially reiterates the prior rejections, albeit with additional remarks. In this regard, this Reply Brief of Appellants incorporates by reference herein the entirety of the previously filed Brief of Appellants. Moreover, additional arguments are also presented below.

A. Arguments directed to the first grounds for rejection: Whether claims 1-3 are obvious under 35 U.S.C. §103(a) over Cheng et al., "Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database," in view of U.S. Patent No. 5,963,936 (Cochrane) and further in view of U.S. Patent No. 6,438,741 (Al-omari).

Appellants' invention, as recited in independent claim 1, is patentable over the combination of Cheng, Cochrane and Al-omari, because independent claim 1 recites limitations not found in the Cheng, Cochrane and Al-omari references.

Nonetheless, the Examiner's Answer states the following (with changes from the Office Action indicated by strikethrough and highlighted in bold):

> 2. Claims **1-3** are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. ("Implementation of Two Semantic Query Optimization Techniques in DB2 Universal Database"), in view of Cochrane et al. (US Patent 5,963,936), and further in view of Al-omari et al. (US Patent 6,438,741).
>
> As to claim 1, Cheng et al. teaches a method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system (see Abstract), the method comprising:
>
> (a) during compilation of the query, maintaining a GROUP BY clause (see Cheng et al. Page 1, Example 1, and Page 5, query 1)
>
> Cheng et al. does not teach with one or more GROUPING SETS, ROLLUP or CUBE operations
>
> Cocharane et al. teaches with one or more GROUPING SETS, ROLLUP or CUBE operations (see column 7, lines 26-30, and column 7, lines 44-48)
>
> Cheng et al. as modified teaches in its original form, instead of rewriting the GROUP BY clause, until after query rewrite (see Cheng et al. Page 1, Example 1, and Page 5, query 1. In Q'1, the group by clause has been retained); and
>
> (b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP, or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets (see Cochrane et al. 8:26-42, Figure 7. This step occurs after the step listed above) comprised of grouping columns (see 11:62-12:15. The GROUP BY sets are comprised of columns a, b, x, and y),
>
> Cheng et al. as modified does not teach generating a query execution plan for the query with a super group block having an array of grouping sets, wherein each pointer points to the grouping sets for a particular one of the levels.
>
> Al-omari et al. teaches generating a query execution plan for the query with a super group block having an array of grouping sets, wherein each pointer points to the grouping sets for a particular one of the levels (see ~~**Figure 3D, 'link**~~

~~mode to GROUP'. Also see~~ 10:36-48, 14:28-35, 41-43, **Also see Figure 15B, 9:31-43 and 16:50-59**);

Cheng et al. as modified teaches:

(c) performing the query execution plan to retrieve data from a database stored on the computer system (see Cochrane et al. 7:41-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Cheng et al. by the teachings of Cochrane et al., since Cochrane et al. teaches that "a method for detecting and stacking grouping sets to support group by operations with grouping sets, rollup, and cube extensions in relational database management systems, with greatly reduced numbers of grouping sets" (see Abstract).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have further modified Cheng et al. by the teachings of Al-omari et al., since Al-omari et al. teaches "a system and method for optimizing complex SQL database queries" (see 3:18-19).

In addition, the Examiner's Answer states the following:

(10) Response to Argument

As to the independent claim, Appellant argues that "the combination of Cheng, Cochrane, and Al-omari does not teach or suggest the limitations of the independent claim, and goes on to state that "the query of Cheng merely illustrates the technique of join elimination in the context of semantic query optimization. However, Cheng merely shows the GROUP BY clause in the same form in both the original query and the optimized query, indicating that the GROUP BY clause is not "maintained during compilation" and then "translated at a later stage of query compilation," as recited in Appellants' claims. Instead, the GROUP BY clause of Cheng is apparently left untouched during the join elimination optimization. Moreover, the GROUP BY clause in Cheng does not have GROUPING SETS, ROLLUP, or CUBE operations in its original form, as admitted by the Office Action. Therefore, the query of Cheng, and the example cited by the Office Action, have no relevance to Appellants' claims".

In response to this argument, it is noted that Cheng et al. teaches wherein two stages of query optimization exist, "query rewrite optimization", and "query plan optimization" (see page 3, section 2, 1st paragraph), and that "after an input query is parsed and converted to an intermediate form called query graph model, the graph is transformed by the Query Rewrite Engine into a logically equivalent but more efficient form using heuristics" (see page 3, section 2, second paragraph). Cheng et al. also states that "The design of the query rewrite engine is ideal for the implementation of SQO" (see page 3, column 2, 1st paragraph after Example 3).

A GROUP BY clause is included in "Query 1" of Cheng et al. (see page 2, Example 1). This "Query 1" is then rewritten using the SQO techniques, to result in Q'1 (see page 5, "Query 1"). "Query 1" contains a "group by" clause. This "group by" clause is maintained while the query is being processed by the "query rewrite engine" using the SQO techniques of Cheng et al., and still exists

unchanged in rewritten query Q'1.

Therefore, the "group by" clause is maintained in its original form, instead of being rewritten, until after query rewrite.

Examiner relied upon Cochrane et al. to teach wherein the query contained one or more GROUPING SETS, ROLLUP, or CUBE operations (see Cochrane et al. 7:26- 30, and 7:44-48).

Appellant argues in regards to Cochrane et al. that "This optimization scheme of Cochrane says nothing about maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP, or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite. Instead, the optimization scheme of Cochrane reduces the GROUP BYs during query rewrite, which means that the GROUP BY clause is not maintained in its original form until after query rewrite, but instead the GROUP BY clause is rewritten". In response to this argument, it is first noted that the rejection was an obviousness rejection in view of Cheng et al., in view of Cochrane et al., and in view of Al-Omari et al., and was based on 35 U.S.C. 103(a).

Cheng et al. first submits a QGM to a "query rewrite engine". Cheng et al. also mentions "query rewrite optimization" and "query plan optimization" (see page 3, column 1). Cochrane et al. receives a query, forms a "query graph model", and rewrites the query, and submit this QGM to a query plan optimizer (see 7:22-35). It is noted that the GROUP BYs in Cochrane et al. are maintained in their original form until the moment of query rewrite. They no longer exist in their original form after query rewrite. Thus, Cochrane et al. does teach "maintaining a GROUP BY clause in its original form until after query rewrite".

In regards to Al-omari et al., Appellants argue that "the groups in Al-omari are in no way equivalent to Appellants' claimed super group block. Specifically, the memo structure of Al-omari includes one or more groups, where each group contains an array of pointers to one or more logical expressions, an array of pointers to one or more physical expressions, an array of pointers to one or more contexts, an array of pointers to one or more plans, and an exploration pass indicator. In Appellants' claims, on the other hand, the super group block supports the translation of a GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, the super group block has an array of pointers, and each pointer of the super group block points to the grouping sets for a particular one of the levels. This super group block of Appellants' claims recites different structure and functions as compared to the memo structure of Al-omari".

In response to this argument, it is noted that the memo structure of Al-omari et al. is a super group block (see 10:36-48, 14:28-35 and 41 -43). The "groups" of the memo structure contain pointers to logical and physical expressions. It is also noted that each group of the memo structure contain pointers to a table expression (see Figure 15B, 9:31-34, and 16:50-59. Table expressions include "group bys". While only "joins" and "scans" are listed in Figure 15B, 9:31-43 and 16:50-59 indicate that "group bys" are also valid table expressions). It is also noted that the Table expressions pointed to by the groups

may accept as input other groups (see 24:38-54). Cochrane et al. teaches to generate a stack of GROUP BYs, in which each previous GROUP BY serves as an input to the next GROUP BY (see 11:34-12:35). As Al-omari et al. teaches a super group block Memo structure with pointers to various table expressions (wherein "GROUP BY" is a valid table expression), wherein each table expression may serve as input to another table expression, and as Cochrane et al. teaches creating a stack of different levels of GROUP BYs, wherein results from GROUP BYs on one level are input into the GROUP BYs on the next, it would have been obvious to one of ordinary skill in the art to have combined the two to have made the currently claimed invention.

Appellants' attorney disagrees with this analysis.

The combination of Cheng, Cochrane and Al-omari does not teach or suggest all the limitations recited in Appellants' independent claim 1. Consider the cited portions of Cheng, Cochrane and Al-omari, which are set forth below (with newly cited portions of Al-omari highlighted in bold):

<u>Cheng: Abstract</u>

In the early 1980's, researchers recognized that semantic information stored in databases as integrity constraints could be used for query optimization. A new set of techniques called semantic query optimization (SQO) was developed. Some of the ideas developed for SQO have been used commercially, but to the best of our knowledge, no extensive implementations of SQO exist today.

In this paper, we describe an implementation of two SQO techniques, Predicate Introduction and Join Elimination, in DB2 Universal Database. We present the implemented algorithms and performance results using the TPCD and APB-1 OLAP benchmarks. Our experiments show that SQO can lead to dramatic query performance improvements. A crucial aspect of our implementation of SQO is the fact that it does not rely on complex integrity constraints (as many previous SQO techniques did); we use only referential integrity constraints and check constraints.

<u>Cheng: Page 2, Example 1</u>

Example 1. Consider the following two queries (both asked against the TPCD [19]). The first query illustrates the technique of Join Elimination.

```
Q1:  select   p_name, p_retailprice, s_name, s_address
     from     tpcd.lineitem, tpcd.partsupp, tpcd.part, tpcd.supplier
     where    p_partkey = ps_partkey and
              s_suppkey = ps_suppkey and
              ps_partkey = l_partkey and
              ps_suppkey = l_suppkey and
              l_shipdate between '1994-01-01' and
              '1996-06-30' and l_discount > 0.1
```

| | |
|---|---|
| group by | p_name, p_retailprice, s_name, s_address |
| order by | p_name, s_name; |

Cheng: Page 5, Query 1

Let the query be $Q_1$ of Example 1. The graphs describing the structure of the joins of the query are shown in Figure 1.

<graphs>

Thus, $Q_1$ can be optimized into $Q_1'$.

| | | |
|---|---|---|
| $Q_1'$: | select | p_name, p_retailprice, s_name, s_address |
| | from | tpcd.lineitem, tpcd.part, tpcd.supplier |
| | where | p_partkey = l_partkey and<br>s_suppkey = l_suppkey and<br>l_shipdate between '1994-01-01' and<br>'1996-06-30' and l_discount > 0.1 |
| | group by | p_name, p_retailprice, s_name, s_address |
| | order by | p_name, s_name; |

Cochrane: Col. 7, Lines 26-30

Generally, the query parser 92 lexes, parses, and semantically checks a query, producing an internal representation (a "query graph model") that is rewritten and submitted to the optimizer which generates an optimized query execution plan.

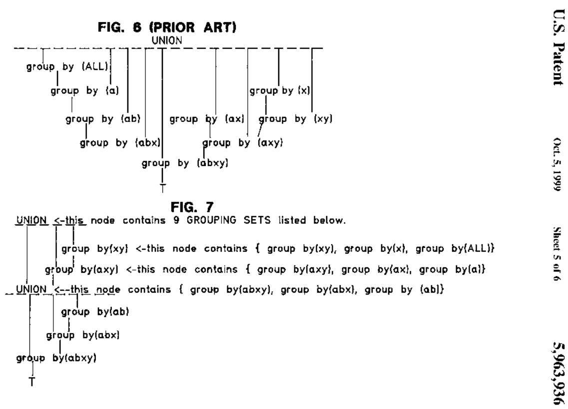
Cochrane: Col. 7, Lines 44-48

The system of FIG. 5 employs the invention to produce a QGM in which the number of GROUP BYs necessary to execute a GROUP BY with multiple GROUPING SETS, concatenated ROLLUPs, or a CUBE has been reduced.

Cochrane: Col. 8, lines 26-42

Now, utilizing the principles of the present invention, and noting the previously derived intersection results shown above at (1)-(4), it becomes possible to construct a query graph model that includes a stacking of GROUP BYs that results in the computation and planning of only 5 GROUP BYs as opposed to the 9 required in FIG. 6. This query graph model is shown in FIG. 7. It should be emphasized that the query graph model of FIG. 7 produces results that are identical to the solution provided in FIG. 6, with only 5 GROUP BY operations, a considerable economy in computational overhead. Indeed, this reduction in the number of GROUP BYs may, in an RDBMS implementing large multi-dimensional tables and subject to complex OLAP queries, be necessary to implement the query. This is due to the fact that the size of such queries, combined with the prior art, can require such large-scale computational assets as to render the query incapable of implementation.

Cochrane: FIGS. 6 and 7




Cochrane: Col. 11, line 62 – col. 12, line 15

As an example, consider the following: GROUP BY ROLLUP(a,b), ROLLUP(x,y) in which the GROUP BY's for ROLLUP(a,b) are:

GROUP BY(a,x,y)

GROUP BY(x,y)

and the GROUP BY's for ROLLUP(x,y) are:

GROUP BY(a,b,x)

GROUP BY(a,b)

Now, the base group for ROLLUP(a,b)ROLLUP(x,y) is determined by base step:

---

UNION1
|||
||GROUP BY(x,y)
|||
|GROUP BY(a,x,y)
||
GROUP BY(a,b,x,y)
|
T

---

Al-omari: FIGS. 3C and 3D

U.S. Patent Aug. 20, 2002 Sheet 4 of 25 US 6,438,741 B1

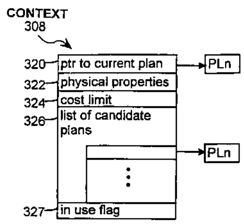



FIG. 3C

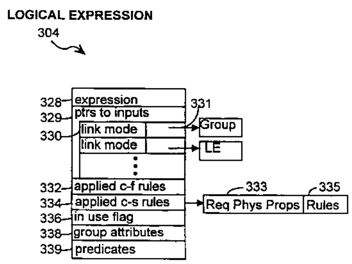



FIG. 3D

Al-omari: col. 10, lines 36-48

Memo: A memo is a search data structure used by the optimizer for representing elements of the search space. The Memo is organized into equivalence classes denoted as groups. Each group includes one or more logical and physical expressions that are semantically equivalent to one another. Expressions are semantically equivalent if they produce the identical output. Initially each logical expression of the input query tree is represented as a separate group in Memo. As the optimizer applies rules to the expressions in the groups, additional equivalent expressions and groups are added. Each group also contains one or more plans and contexts. A context represents plans having the same optimization goal.

Al-omari: col. 14, lines 26-35 and lines 41-43 (actually lines 26-50)

Referring to FIGS. 3A-3E, the Memo 122 includes one or more groups 302, where each group 302 contains an array of pointers to one or more logical expressions 304, an array of pointers to one or more physical expressions 306, an array of pointers to one or more contexts 308, an array of pointers to one or more plans 305, and an exploration pass indicator 307. A logical expression, physical expression, context, and plan are described in more detail below. An exploration pass indicator 307 indicates for each pass whether or not the group has been explored. Preferably, the exploration pass indicator is a bitmap having n bits with one or more bits representing a particular pass and indicating whether or not exploration was performed in the pass.

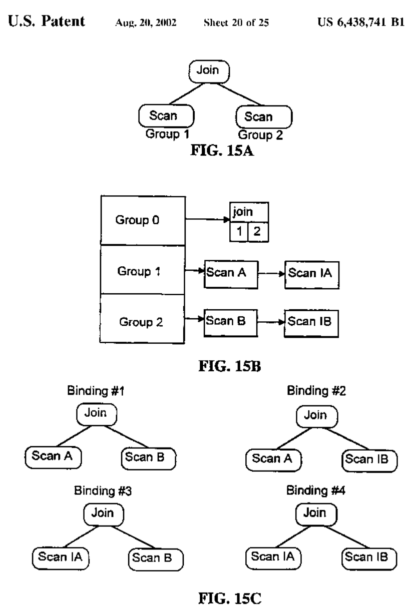
Each logical expression 304 is represented as a data structure that stores the particular expression 328 and has pointers 331 associated with each input

expression 329. Each pointer 331 has a link mode 330 that specifics the datum that the pointer addresses. Preferably, there are two link modes associated with an input expression: a memo mode and a binding mode. In memo mode, the pointer 331 identifies the group corresponding to the input expression. In binding mode, the pointer 331 identifies a logical expression that is part of a binding.

Al-omari: col. 3, lines 18-19

The invention is a system and method for optimizing complex SQL database queries.

**Al-omari: FIG. 15B**




**Al-omari: col. 9, lines 31-43**

**Table Expression: A table expression is a relational expression that produces a table or set of rows. Examples of table expression operators include Scan (or Retrieve), Join, Union, and Group By.**

**Logical Operator: A logical operator represents an implementation-independent operation (e.g., join or scan).**

**Physical Operator: A physical operator specifies a particular implementation method or procedure (e.g., hashjoin, mergejoin, etc.).**

**Expression tree: An expression tree corresponds to a relational expression having one or more logical or physical expressions. The expression tree includes one or more nodes, each node is classified as a logical expression or a physical expression. Each node can contain zero or more inputs, each input being a relational expression. The expression tree includes one or more levels, each level containing nodes that are inputs to a node of a**

**preceding level. The root node represents a relational expression having the top-most operator and positioned in the first level.**

**Al-omari: col. 16, lines 50-59**

**Thus, each operator of each expression has the following characteristic inputs and outputs:**

| **Operator** | **Group Attributes Characteristic Inputs** | **Characteristic Outputs** |
|---|---|---|
| **GroupBy 10** | **10** | **Results of the query** |
| **NestedJoin** | **10** | **A.y, B.x** |
| **ScanA** | **10** | **A.y, A.z** |
| **ScanB** | **A.z** | **B.x** |

Appellants' attorney respectfully submits that the combination of Cheng, Cochrane and Al-omari does not teach or suggest Appellants' independent claim 1, which is set forth below:

> 1. A method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system, the method comprising:
>
> (a) during compilation of the query, maintaining a GROUP BY clause with one or more GROUPING SETS, ROLLUP or CUBE operations in its original form, instead of rewriting the GROUP BY clause, until after query rewrite;
>
> (b) at a later stage of query compilation, translating the GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, wherein each of the levels has one or more grouping sets comprised of grouping columns, and generating a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels; and
>
> (c) performing the query execution plan to retrieve data from a database stored on the computer system.

Cheng shows the same GROUP BY clause in both the original query $Q_1$ (on page 2 of Cheng) and the optimized query $Q_1'$ (on page 5 of Cheng). This indicates that the GROUP BY clause is not "maintained during compilation" and then "translated at a later stage of query compilation," as recited in Appellants' claims.

Moreover, the GROUP BY clause in Cheng does not have GROUPING SETS, ROLLUP or CUBE operations. The fact that the GROUP BY clause of Cochrane does have GROUPING SETS, ROLLUP or CUBE operations does not overcome the deficiencies of Cheng, because, like Cheng, the GROUP BY clause of Cochrane is not "maintained during compilation" and then

"translated at a later stage of query compilation," as recited in Appellants' claims. Instead, the optimization scheme of Cochrane reduces the GROUP BYs during query rewrite, which means that the GROUP BY clause is rewritten.

Finally, the memo structure of Al-omari is not "a super block structure." Instead, the memo structure of Al-omari is merely an array of one or more groups, where each group contains an array of pointers to one or more logical expressions, an array of pointers to one or more physical expressions, an array of pointers to one or more contexts, an array of pointers to one or more plans, and an exploration pass indicator. For example, in Figure 15B, Al-omari shows logical expressions for Group 0 (join 1,2), Group 1 (Scan A, Scan 1A) and Group 2 (Scan B, Scan 1B).

The mere fact that Al-omari refers to his memo structure as having "groups" does not alter the fact that there is nothing in the memo structure of Al-omari that teaches or suggests any of the remaining portions of Appellants' claim limitations, i.e., Al-omari does not translate a GROUP BY clause with the GROUPING SETS, ROLLUP or CUBE operations into a plurality of levels, Al-omari does not teach that each of the levels has one or more grouping sets comprised of grouping columns, and Al-omari does not generate a query execution plan for the query with a super group block having an array of pointers, wherein each pointer points to the grouping sets for a particular one of the levels. Consequently, the memo structure and its groups in Al-omari are in no way equivalent to Appellants' claimed super group block.

Thus, Appellants' attorney submits that independent claim 1 is allowable over the combination of Cheng, Cochrane and Al-omari, because the references, taken individually or in combination, fail to teach or suggest the limitations of Appellants' claimed invention. Further, dependent claims 2 and 3 are submitted to be allowable over the combination of Cheng, Cochrane and Al-omari in the same manner, because they are dependent on independent claim 1, and thus contain all the limitations of the independent claim. In addition, dependent claims 2 and 3 recite additional novel elements not shown by the combination of Cheng, Cochrane and Al-omari, as set forth in more detail in the Brief of Appellants.

## III. CONCLUSION

In light of the above arguments, Appellant's attorney respectfully submits that the cited references do not anticipate nor render obvious the claimed invention. More specifically,

Appellant's claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103.

As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Appellants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: June 1, 2009

By: /George H. Gates/
Name: George H. Gates
Reg. No.: 33,500

GHG/